

Dec. 22, 2020
@2020 SWELL

PsychoPy Tutorial

Building a simple experiment and deploying online

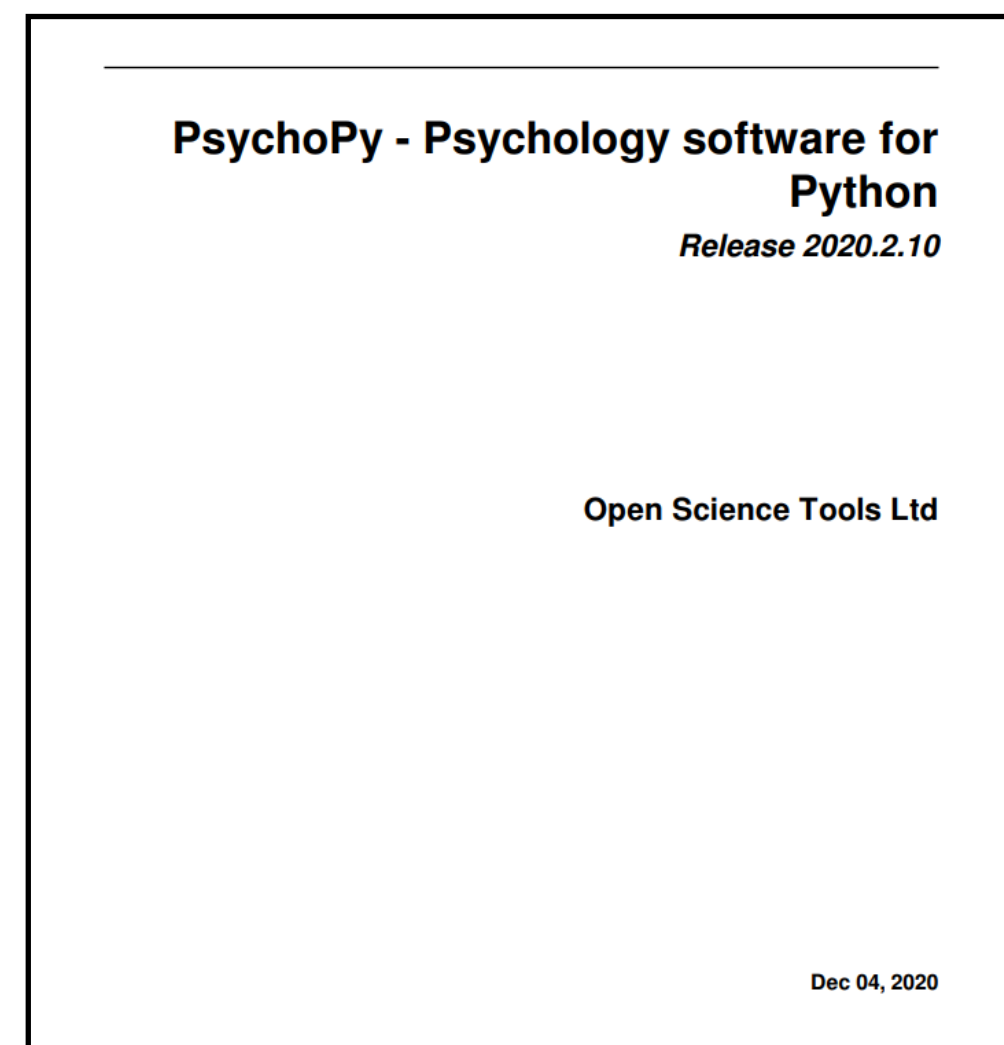
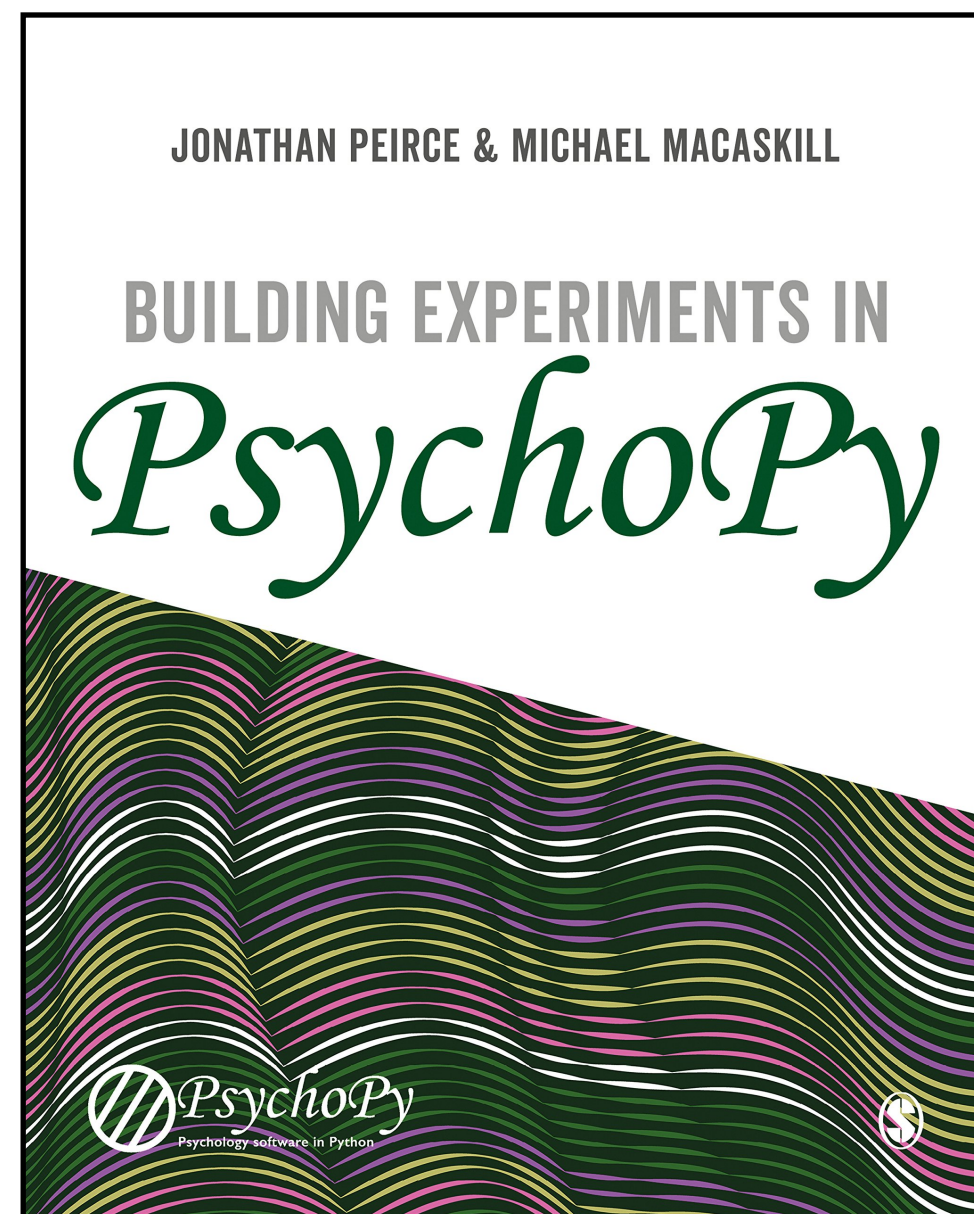
Gyuhwan Lee (popcorn426@snu.ac.kr)
Meanings & Melodies Lab

PsychoPy ?

- “PsychoPy is an application for the creation of experiments in behavioral science (psychology, neuroscience, linguistics, etc.) with precise spatial control and timing of stimuli. ... users can write scripts in Python if they choose, while those who prefer to construct experiments graphically can use the new Builder interface.” (Peirce et al. 2019)
- “new opportunities for psycholinguistic research that are made available by presenting experiments online over the web. We focus on PsychoPy3, which is a new version of a system for the development and delivery of behavioural experiments. Crucially, it allows for both these functions to be performed online.” (Gallant & Libben 2019)

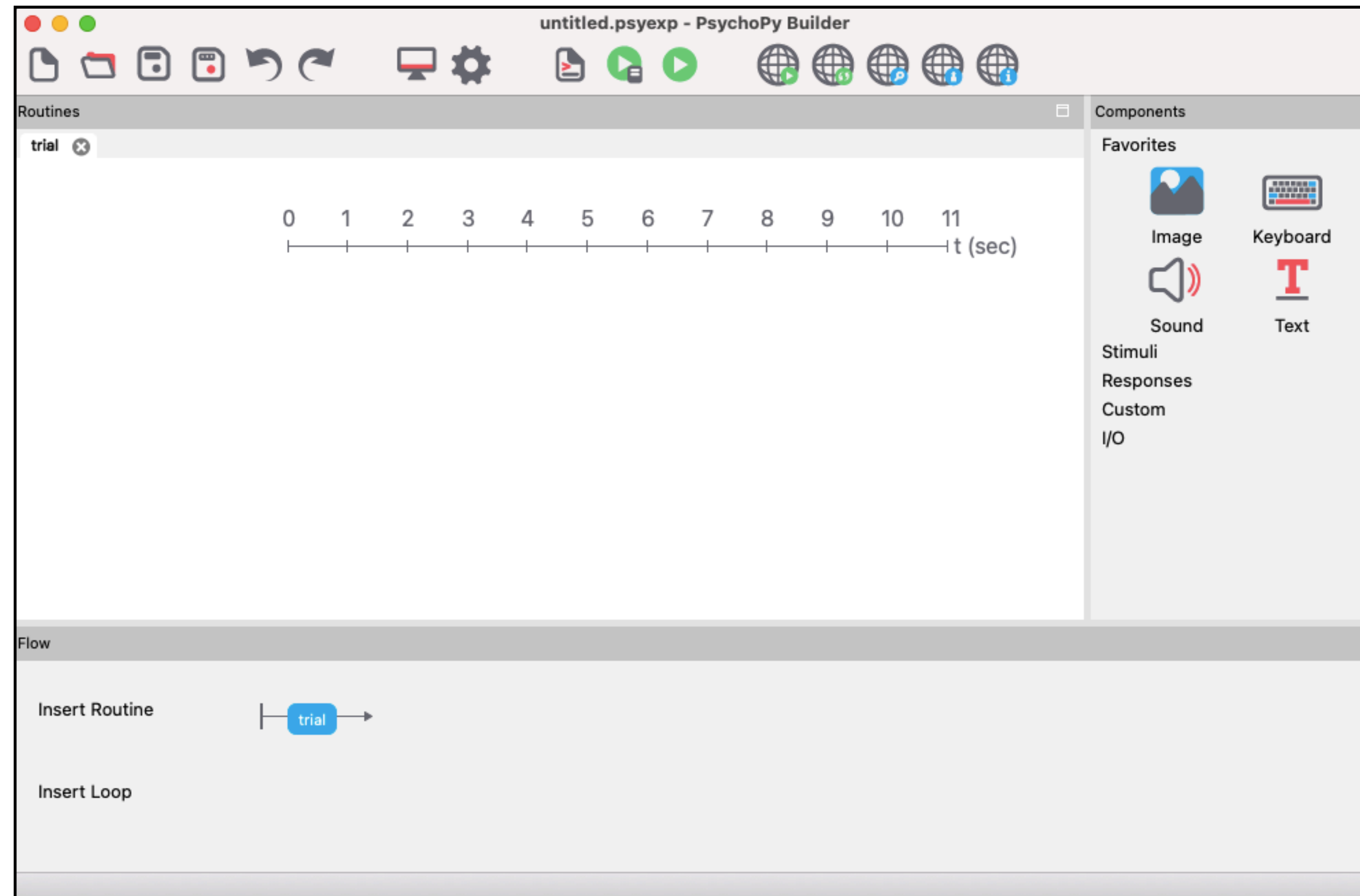
Reference Materials

- Building Experiments in PsychoPy (2018)
- PsychoPy - Psychology software for Python - Release 2020.2.10
(<https://www.psychopy.org/PsychoPyManual.pdf>)



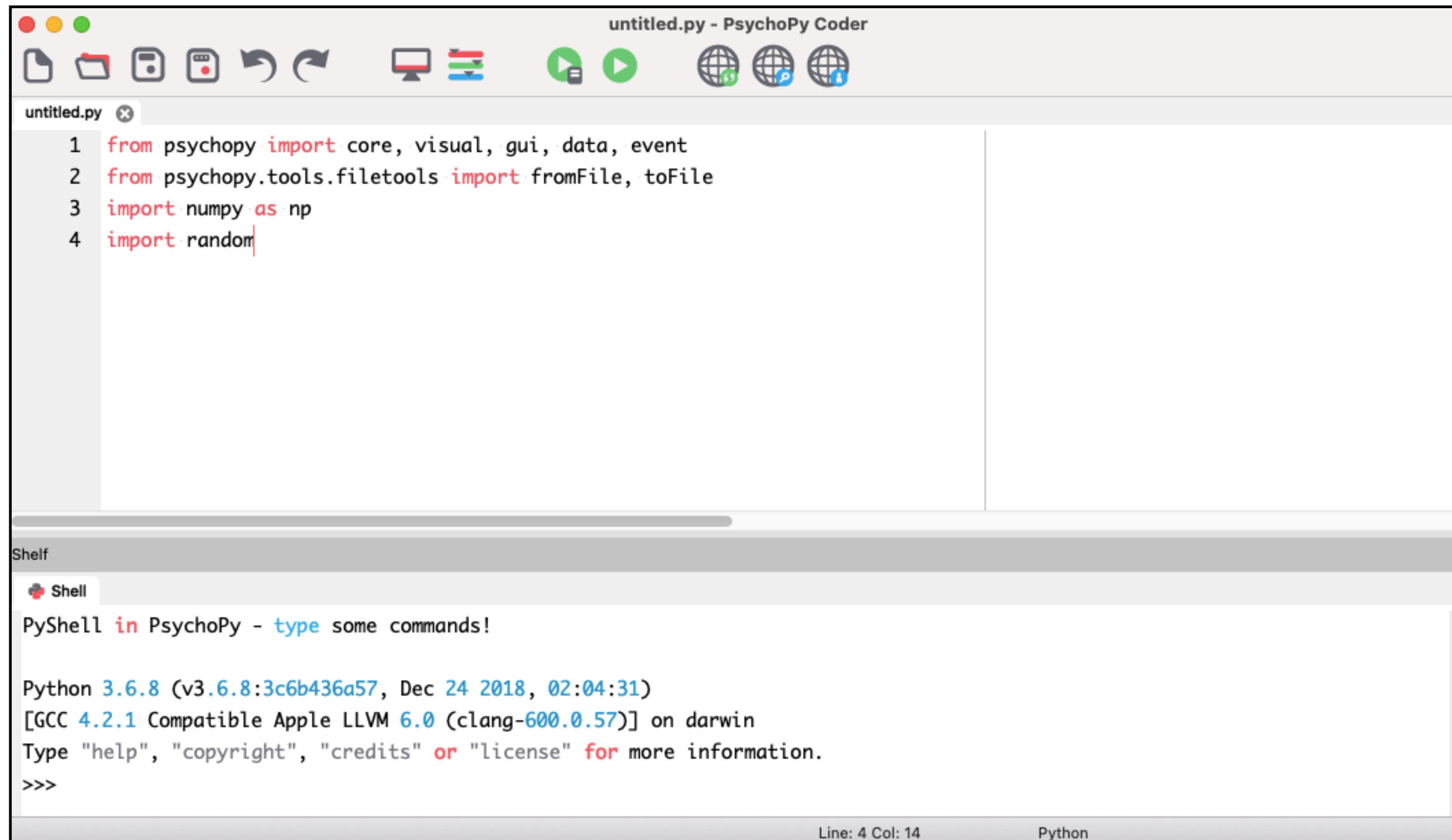
Components of PsychoPy

- **Builder** + Coder + Experiment Runner



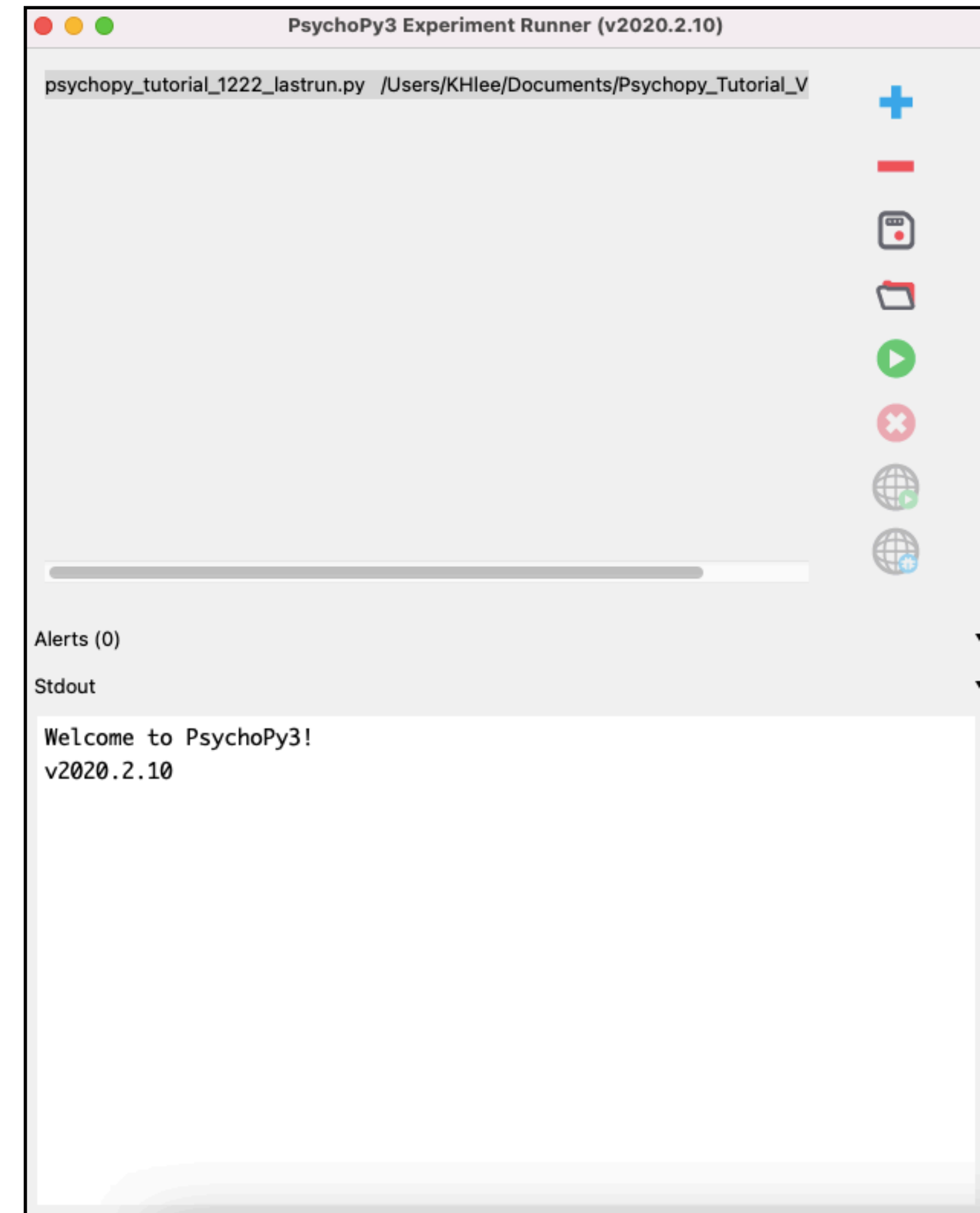
Components of PsychoPy

- Builder + **Coder** + Experiment Runner



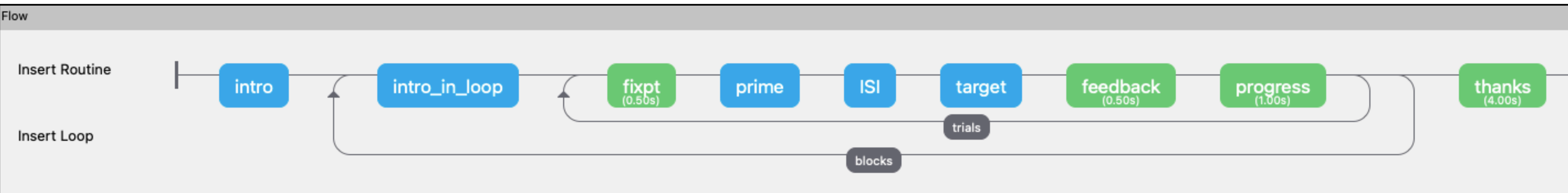
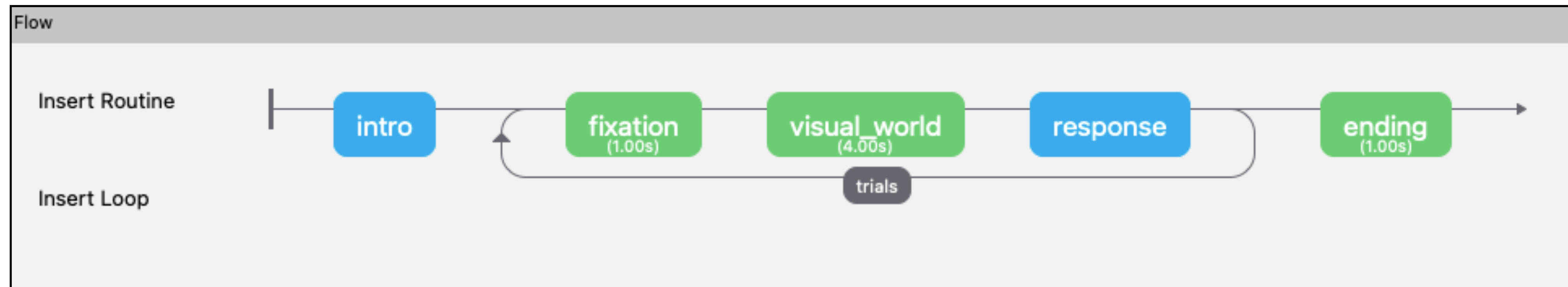
Components of PsychoPy

- Builder + Coder + **Experiment Runner**



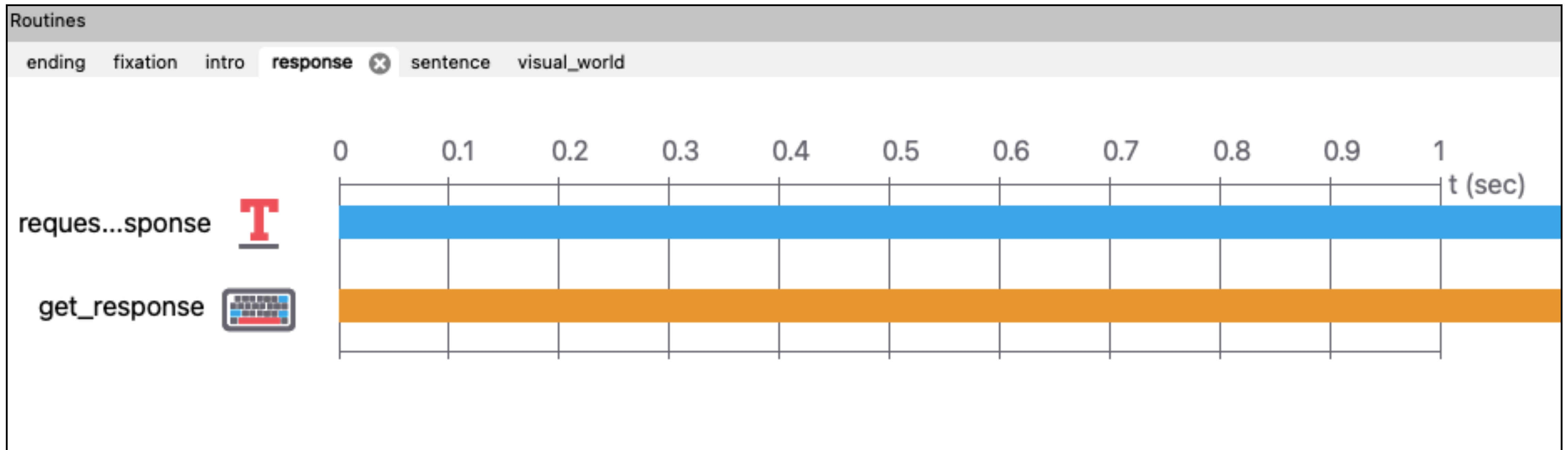
Elements of PsychoPy Builder

- **Flow** ⊃ Routines & Loops ⊃ Components



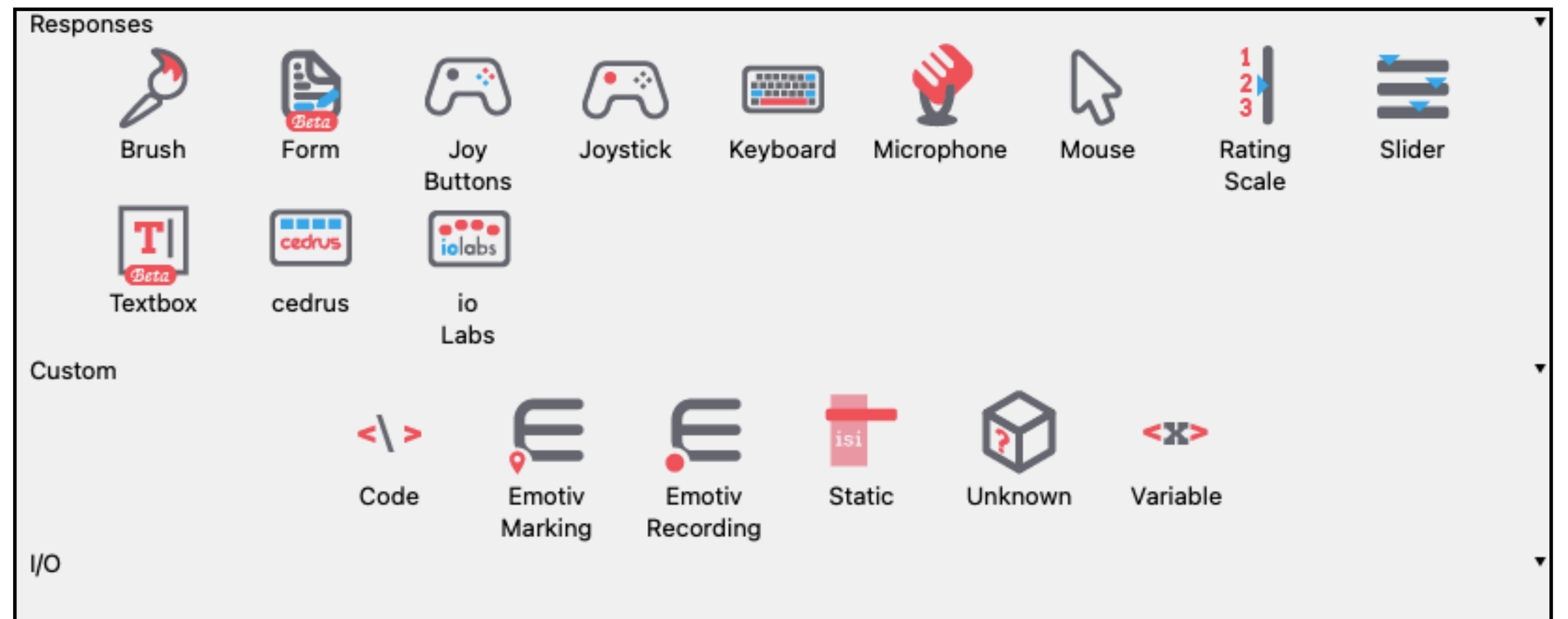
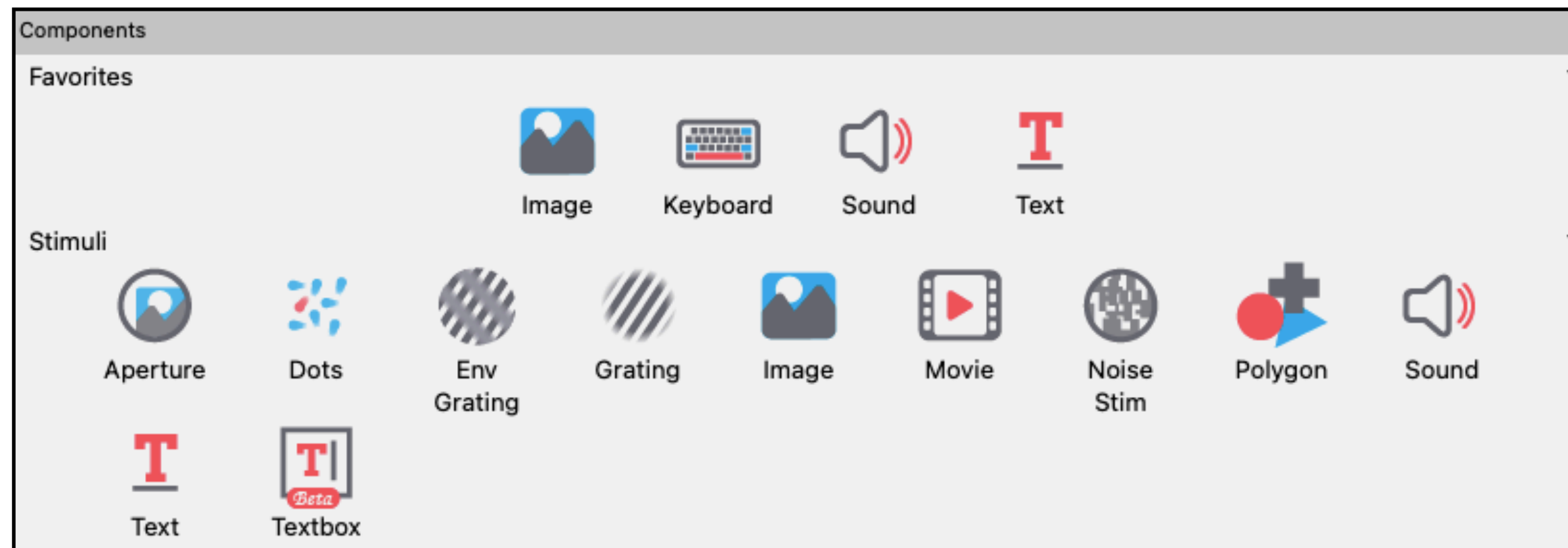
Elements of PsychoPy Builder

- Flow \supset **Routines** & Loops \supset Components



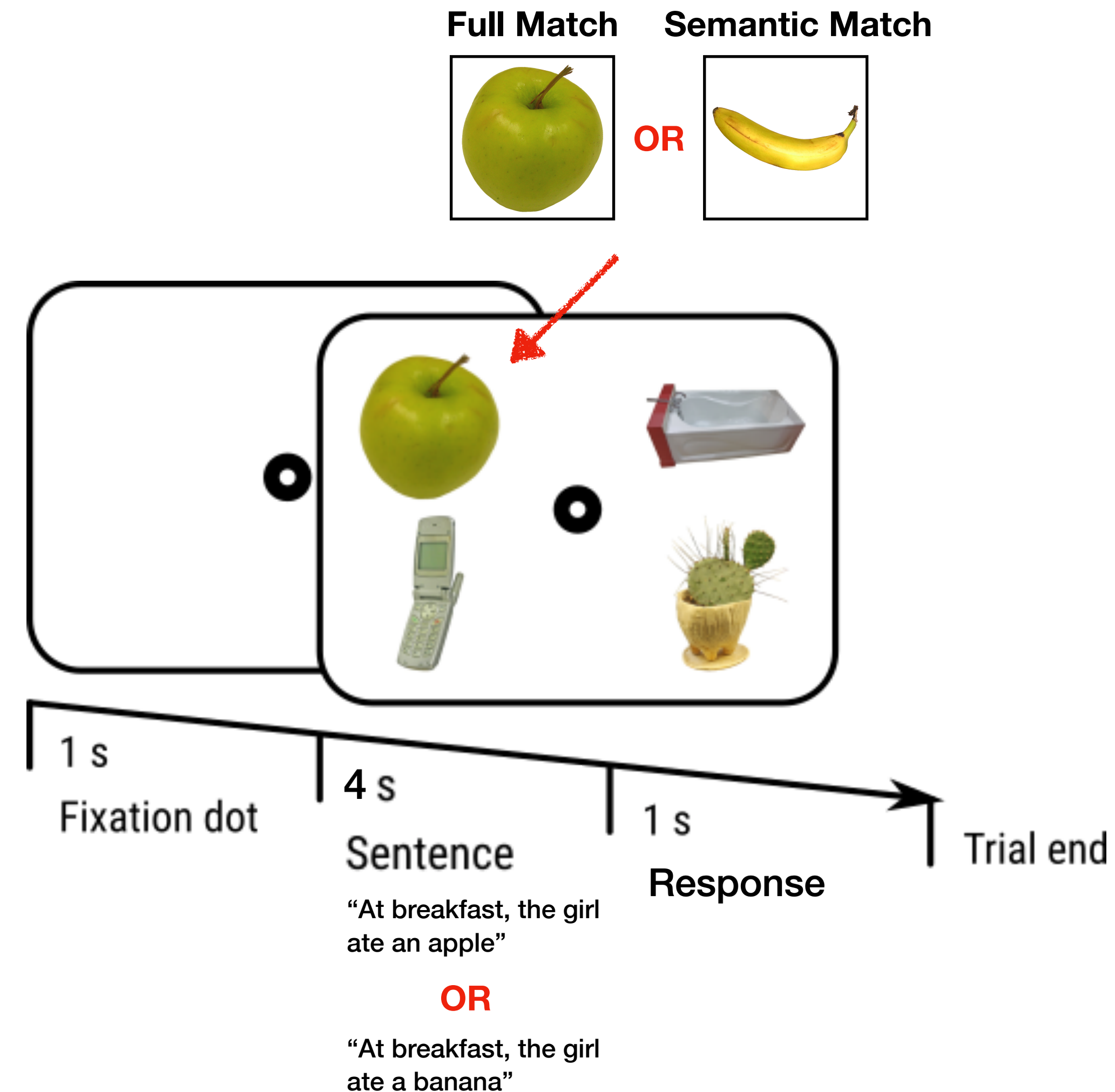
Elements of PsychoPy Builder

- Flow ⊃ Routines & Loops ⊃ **Components**



Demo experiment we will implement

- Inspired by — <https://osdoc.cogsci.nl/3.3/tutorials/visual-world/>
- Visual world paradigm
- 1 target object + 3 distractors
- One factor (Target Match) with two levels (Full or Semantic), varied within subjects.
- 16 recorded sentences, 16 target objects
- Every sentence and every target object is shown twice: once in the Full Match condition, and once in the Semantic Match condition
- Stimuli (images & sounds) can be downloaded from — <https://osf.io/z27rt/>

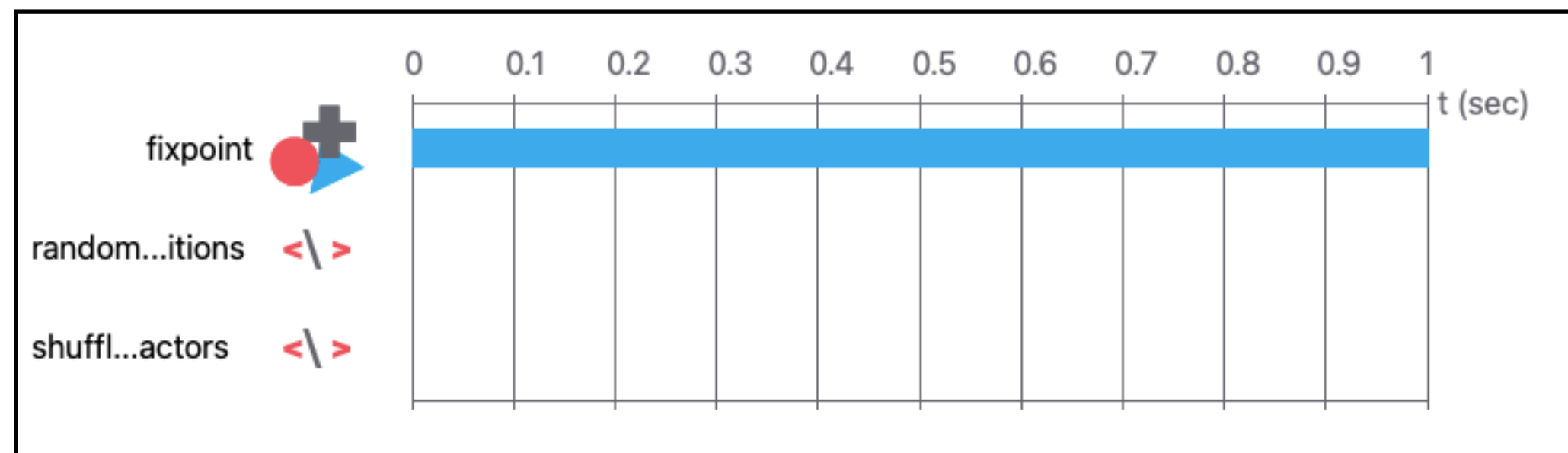


Demo experiment implemented in the Builder



Utilizing custom codes

- Code components can be added
 - You can select when the block is executed (before experiment, begin routine etc.)
 - Code can be written in two languages (Python & Javascript)
 - JS -> Py auto-conversion is supported (but not perfect)



- Eg. for randomizing image positions

```
randomize_positions Properties
Name: randomize_pos Code Type: Both

Before Experiment:
1 lb = (-0.4, -0.3)
2 lt = (-0.4, 0.3)
3 rb = (0.4, -0.3)
4 rt = (0.4, 0.3)
5
6 positions = [lb, lt, rb, rt]
7
8 random_permute = np.random.permutation(4)
9

Each Frame:
1 lb = [(-0.4), (-0.3)];
2 lt = [(-0.4), 0.3];
3 rb = [0.4, (-0.3)];
4 rt = [0.4, 0.3];
5
6 positions = [lb, lt, rb, rt];
7
8 random_permute = [0, 1, 2, 3].sort()
9
```

- Eg. for presenting random distractors

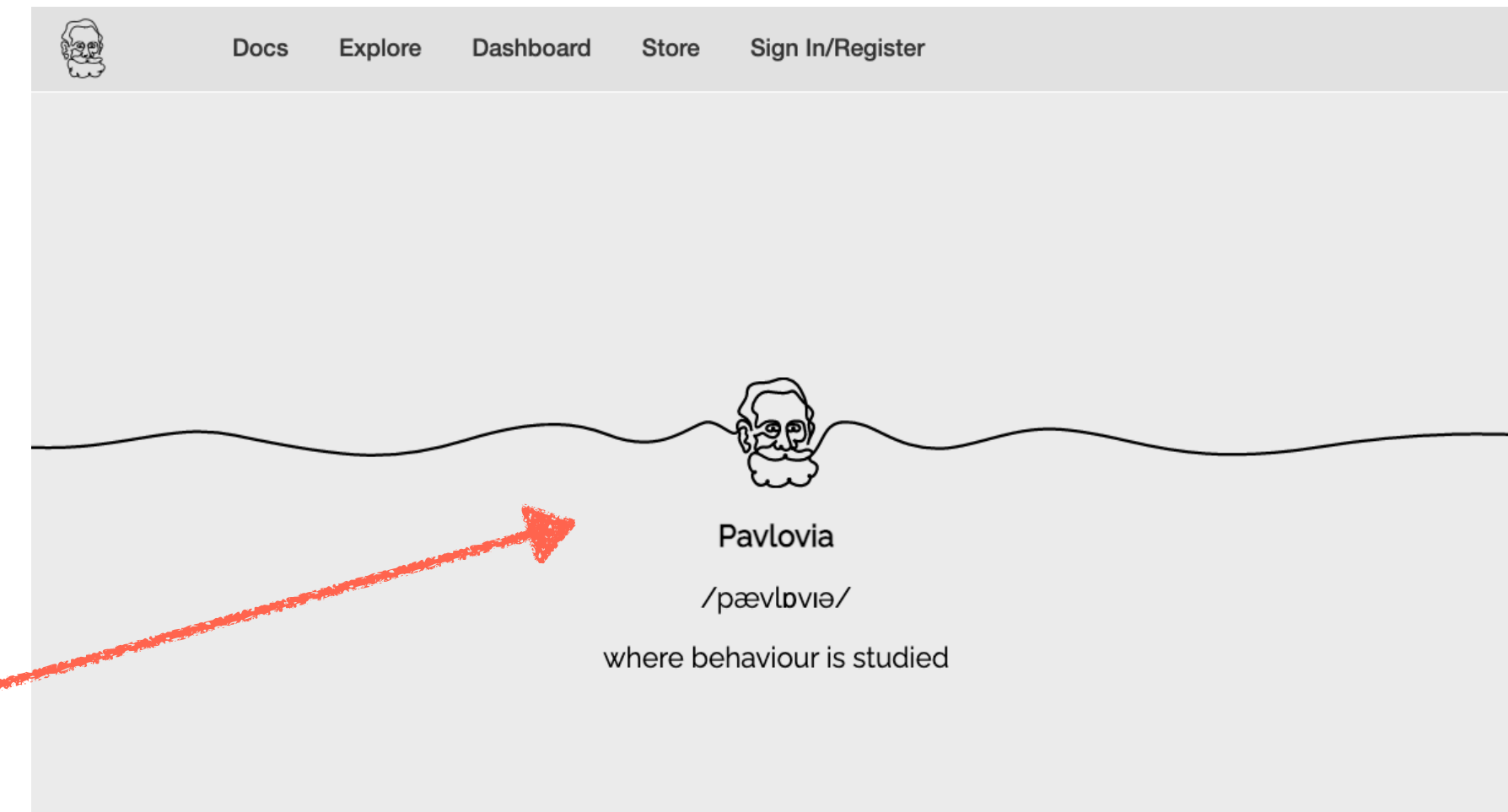
```
shuffle_distractors Properties
Name: shuffle_distract Code Type: Both

Before Experiment:
1 selected_distractors = np.random.random(4)
2
3 pic2 = distractor_images[selected_distractors[0]]
4 pic3 = distractor_images[selected_distractors[1]]
5 pic4 = distractor_images[selected_distractors[2]]
6
7 thisExp.addData('picture2', pic2)
8 thisExp.addData('picture3', pic3)
9 thisExp.addData('picture4', pic4)

Each Frame:
1 pic2 = distractor_images[Math.floor(Math.random() * 4)]
2 pic3 = distractor_images[Math.floor(Math.random() * 4)]
3 pic4 = distractor_images[Math.floor(Math.random() * 4)]
4 thisExp.addData("picture2", pic2);
5 thisExp.addData("picture3", pic3);
6 thisExp.addData("picture4", pic4);
7
```


Going ONLINE

- Now it's time to use these icons



1. Go to Pavlovia (<https://pavlovia.org/>), create an account
2. Clicking one of the first two icons will auto-convert experiment script in .py (Python) to .js (Javascript) and ask you to create a repository (Pavlovia runs based on Gitlab version control system)
3. Upload your experiment by syncing your local folder with the online repository
4. When your experiment is up, you can assign credits to the experiment and start sharing the experiment link (you need to buy credits from Pavlovia, but it really doesn't cost much)

Going ONLINE

Your experiment is now online

pa

pavlovia_tutorial_visual_world

pavlovia_tutorial

★ 40 4.0 RUNNING

created: 2020-12-19 updated: 2020-12-22

GitLab id: #90248

Inspired by - <https://osdoc.cogsci.nl/3.3/tutorials/visual-world/>

Keywords:

Click to add or edit keywords

Software Platform

Platform Version

PSYCHOJS

2020.2

View code

Pilot

Run

Status

INACTIVE

PILOTING

RUNNING

The experiment is available on the Pavlovia server but cannot be run. Change its status to PILOTING to test it, and RUNNING to make it available to participants.

You can pilot the experiment to test that it is working adequately. Pressing the [Pilot] button (above) generates a new URL, which is valid for 1h each time.

Participants can run the experiment, provided that they meet the constraints of its recruitment policy and that either enough credits or a valid license are available.

Running Mode

CREDIT

LICENSE

You do not have enough credits assigned to this experiment to run it by consuming them. You can pilot your

Recruitment

Policy

Constraints

URL

Change

Anyone meeting the experiment constraints can run it by opening the experiment Url, provided that either enough credits or a valid license are available.

Url: https://run.pavlovio.org/pavlovia_tutorial/pavlovia_tutorial_visual_world

Saving Results

Format

Incomplete Results

CSV

DATABASE

Experiment results are saved in .csv files, which are stored on the Pavlovia server and in the experiment's GitLab repository.

You can retrieve them by pressing the [Download results] button, or directly from the repository, or by git pulling from your local repository.

Save incomplete results

All results will be saved, even when the participant does not complete the experiment.

If the experiment's running mode is CREDIT, a credit will be consumed at the end of every session, even those prematurely interrupted.

This is an experimental feature currently available for:

- PsychoPy/PsychJS: 'unversioned', 3.2, 3.2.0, 3.2.2, 2020.1, and 2020.2
- jsPsych: all versions
- labjs: all versions

Assigning credits

Status

INACTIVE

PILOTING

RUNNING

The experiment is available on the Pavlovia server but cannot be run. Change its status to PILOTING to test it, and RUNNING to make it available to participants.

You can pilot the experiment to test that it is working adequately. Pressing the [Pilot] button (above) generates a new URL.

Participants can run the experiment, provided that they meet the constraints of its recruitment policy and that either enough credits or a valid license are available.

Running Mode

CREDIT

LICENSE

You do not have enough credits assigned to this experiment to run it by consuming them. You can pilot your experiment but participants will not be able to run it until you assign credits to your experiment or your institution obtains a license.

0

60

23

assigned

available

reserved

consumed

0

60

0

23

Update assigned credits

Release reserved credits

An experiment can be in three states: inactive, piloting, and running 1 participant = 1 credit

Folder structure when we're done

Psychopy_Tutorial_Visual_World				
이름	수정일	크기	종류	
> data	오늘 오전 4:14	--	폴더	
index.html	그저께 오후 4:57	2KB	HTML 텍스트	
psychopy_tutorial_1222_lastrun.py	오늘 오전 4:14	37KB	Python 스크립트	
psychopy_tutorial_1222-legacy-browsers.js	그저께 오후 4:57	42KB	JavaScript	
psychopy_tutorial_1222.js	오늘 오전 3:49	41KB	JavaScript	
psychopy_tutorial_1222.js.bk	그저께 오후 5:42	41KB	JavaScript	
psychopy_tutorial_1222.psyexp	오늘 오전 4:15	52KB	PsychoPy 문서	
psychopy_tutorial_1222.py	그저께 오후 4:23	37KB	Python 스크립트	
> resources	2020년 12월 17일 오전 2:20	--	폴더	
stimuli_set	오늘 오전 4:36	4KB	탭으로 분리된 값	
stimuli_set_simple	그저께 오후 2:28	2KB	CSV 문서	

- You can download the .zip file of this folder at — <https://drive.google.com/file/d/1rER272MXw-yAyAQcBihn4C1tY0ItukhV/view?usp=sharing>

Reference

- Gallant, J., & Libben, G. (2019). No lab, no problem: Designing lexical comprehension and production experiments using PsychoPy3. *The Mental Lexicon*, 14(1), 152-168.
- Peirce, J., & MacAskill, M. (2018). *Building experiments in PsychoPy*. Sage.
- Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., ... & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior research methods*, 51(1), 195-203.
- Peirce, J., (2020) PsychoPy - Psychology software for Python - Release 2020.2.10 (<https://www.psychopy.org/PsychoPyManual.pdf>)